Detecting and Classifying Sexism by Ensembling Transformers Models

Alejandro Vaca-Serrano[1]

[1]Instituto de Ingeniería del Conocimiento, Francisco Tomás y Valiente st., 11 EPS, B Building, 5th florr UAM Cantoblanco. 28049 Madrid, Spain

Abstract

This work presents the system with the highest results in terms of f1-score for tasks 1 and 2 of EXIST2022. It is a challenge formed of two tasks, aimed at identifying and categorizing sexism in texts, respectively. First of all, a review of language models in Spanish and English is carried out, identifying the best performing models in each language. Also, a review of similar tasks and challenges (sexism detection in texts) is done. Then, models are trained in two phases. The first phase is for selecting the best hyperparameters for each model, while in the second phase these hyperparameters are used to learn with more training data. Finally, a simple ensembling strategy is used, which takes into account the performance of each model over a small validation set. This is compared against building a pure Transformers Ensemble, showing that the simple ensembling strategy obtains higher results. This leaves for future work the task of making such Ensembles work at least as good as the naive ensembling strategy.

Keywords

Transformers, Ensemble, Sexism Detection, Sexism Categorization

1. Introduction

In this work, we explore different Transformer-based solutions for two subtasks of EXIST2022 (sEXism Identification in Social neTworks) [1], as part of Iberlef2022. Apart from single models, some ensembling strategies are tried and reported. This event has the objective to promote the research on NLP tools for detecting sexism in texts both in Spanish and English.

First of all, previous related work is reviewed in section 2, then, tasks are described in section 3. Models are presented in section 4, together with their evaluation results. Section 5 deals with experiments carried out to build pure transformers ensemble models, while section 6 presents the results in the final test set of the competition. Finally, in section 7, conclusions and future work are presented.

2. Related Work

There has been an increasing interest towards sexism detection tasks in the recent years. One example of such effort is [2], where a sexism classification dataset is presented in Spanish and

IberLEF 2022, September 2022, A Coruña, Spain.

alejandro_vaca0@hotmail.com (A. Vaca-Serrano)
https://www.linkedin.com/in/alejandro-vaca-serrano/ (A. Vaca-Serrano)

English, proposing solutions based on n-grams and classic machine learning models. Using the English part of that same dataset and combining it with other similar ones, [3] followed a similar approach of using classic Machine Learning to detect sexism and misogyny. There have been more recent approaches to the same task. For instance, [4] obtains State-of-the-Art (SoTA) results detecting sexism in the workplace by using GloVe Embeddings [5] and modified LSTMs, adding attention mechanisms to them [6].

As this is the second edition of EXIST, there are several works from 2021 edition dealing with both sexism binary classification and sexism categorization. Some examples are [7]. In that work, multilingual Transformer models are used for both tasks. In 2021, the winning team for both tasks was [8]. In [8], authors explain how they used multilingual and monolingual models, together with ensemble models. Finally, [9] summarizes all works presented for EXIST21, together with their results. Other works dealing with sexism detection are [10] and [11]. The second one is based on multilingual Transformers models, which are bigger and trained with more data than the existing Spanish-only models. However, models specific to a language tend to perform better in tasks specific of that language.

2.1. Language Models in Spanish and English

As both tasks 1 and 2 have texts in Spanish and English, the State-of-the-Art of both models in terms of language models are reviewed, justifying the further models selection for both languages.

2.1.1. English

The language that has the most number and quality of language models is English, with no doubt. Since the release of BERT [12], many language models have been released in English. Some of the most remarkable are arguably RoBERTa [13], T5 [14] or DeBERTa [15]. For the EXIST tasks, encoder-based models such as RoBERTa or DeBERTa are the most interesting, as they tend to work better than decoder-based or encoder-decoder models for Natural Language Understanding (NLU) tasks.

More recently, a new version of DeBERTa, DeBERTa v3, was released [16]. In [16] it is shown that DeBERTa v3 improves over DeBERTa [15] on several tasks. DeBERTa, on the other hand, clearly outperforms RoBERTa and BERT on a series of benchmarks [15].

For this reason, the first model chosen for English is the large version of DeBERTa v3. Although RoBERTa is supposed to work slightly worse than DeBERTa, it is a very commonly used model which typically provides good results; in particular, it tends to perform better than BERT [13]. Therefore, RoBERTa large is also used for English texts.

These two models, RoBERTa and DeBERTa v3, are generalist models, that is, they have been trained with a general domain corpus. However, this task is from a very concrete domain: the social networks domain; therefore it is desirable that also a domain-specific model is used for this task. BERTweet [17] is a Twitter version of RoBERTa, trained solely with Twitter data. In [17] it is shown that it performs generally better than generalist models for Twitter-domain specific tasks. For this reason, it was decided to use BERTweet-large for the English texts.

Finally, models RoBERTa-large, BERTweet-large and DeBERTa v3-large are used for English.

Although there are many more language models available in English, it was preferred to train more large models in this case, at the expense of training less varied models.

2.1.2. Spanish

In Spanish there are not so many language models, although in the last year some have been released. The first language model released in Spanish was BETO [18], a Spanish BERT. Then, in the context of the MarIA project [19], Spanish RoBERTa and GPT-2 models were released, both base and large. As we are only interested in encoder-based models for this work, due to the nature of EXIST tasks, RoBERTa-base model will be referred to as MarIA-base, while RoBERTa-large will be called MarIA-large. When models were trained for EXIST, MarIA-large had still some convergence issues, and the final stable version that is available today, was not available. For that reason it was discarded, although in the last version of [19] it is shown that it performs currently better than its base counterpart. Finally, BERTIN model was released this year [20]. It is also a version of RoBERTa in Spanish, trained with less resources than [19] but with novel techniques.

As with English, we find it desirable to use a Twitter-specific language model in Spanish. In this regard, RoBERTuito [21] is a Spanish version of RoBERTa trained with twitter data only. Finally, models BETO [18], BERTIN [20], MarIA-base [19] and RoBERTuito [21] are chosen for Spanish. One more model is used for Spanish than for English, as for the latter more large models are available, therefore we compensate for the smaller models in Spanish by adding more models.

3. Tasks Description

3.1. Task 1

The first task consists on a binary classification task, that is, systems must decide whether a tweet is sexist or not. Figure 1 represents label distribution in Spanish, while figure 2 is for the English part of the dataset. Both figures only take into account the train split of the data.

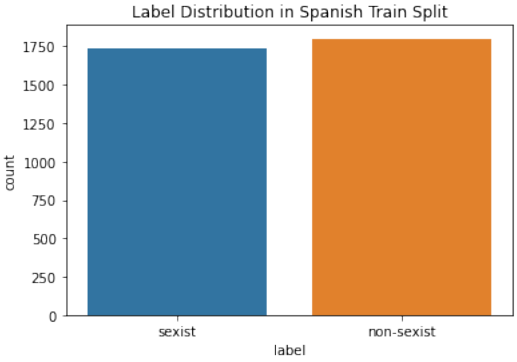

Figure 1: Number of elements per class for task 1 in Spanish.

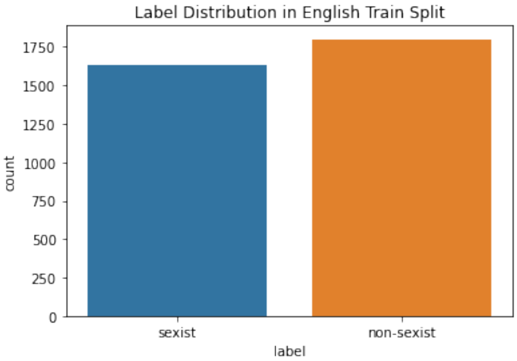

Figure 2: Number of elements per class for task 1 in English.

It is clear from the figures that English data is more unbalanced than Spanish data. However, in both cases there is a good balance between positive and negative labels, as there is not much difference between the proportions of each of them.

3.2. Task 2

On the other hand, for task 2 there are many more labels. Concretely, labels for this task are: ideological and inequality, stereotyping and dominance, objectification, sexual violence, misogyny and non-sexual violence and non-sexism. Due to the little training data and the number of labels, for this task the labels matrix is very sparse, meaning there are relatively few examples of each label, therefore models are expected to find it harder to complete this task. It is determined that each tweet must correspond to one label, that is, a text cannot be simultaneously categorized into more than one type of sexism. For this reason the task is modelled as a multiclass classification task, and not as a multilabel one.

Figures 3 and 4 represent the number of elements of each class for task 2 in each of the languages. As can be seen in the images, in both languages labels are clearly unbalanced, with non-sexist label having much more elements than the rest. This makes sense, since the data is the same as in task 1, therefore around half of the tweets, labelled in task 1 as sexist, are splitted between 5 classes. The distribution for the sexist labels is more or less uniform, with any of the labels being too underrepresented.

4. Models Training

The base models for this section were discussed and explained in 2.1. In this section the training procedure is explained.

For both tasks a similar approach was used. Training was splitted in two different phases, plus a mixing phase without training, one building on top of the previous one.

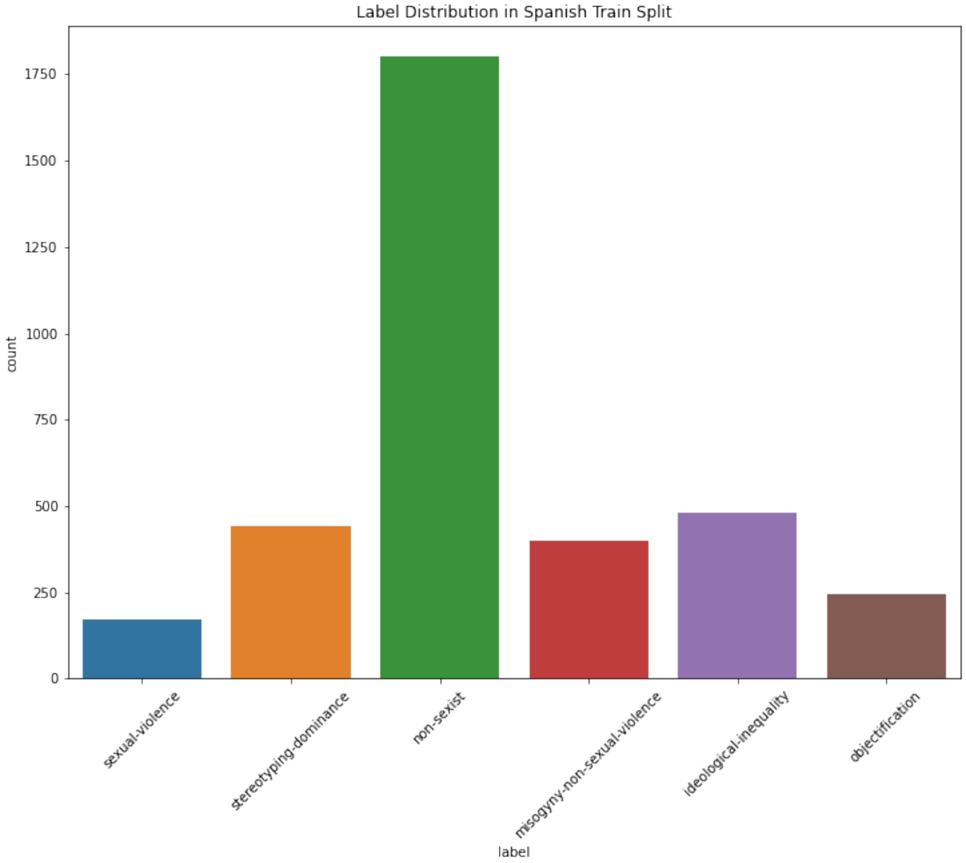

Figure 3: Number of elements per class for task 2 in Spanish.

4.1. Phase 1: Hyperparameter Optimization

First of all, we use the given train and validation splits to find the best hyperparameters for each of the models used. As the validation data corresponds to the test data from past year, this enables us to measure the resulting models against past years' results.

For doing this, Optuna [22] was used, together with Huggingface Transformers [23]. Given the low volume of texts for both tasks, experiments were carried out such that 70 trials in total were run per model. Out of those 70, 25 were random initial trials, while the 45 next trials were optimized by Optuna.

Table 1 shows the hyperparameter space used for this first training step. The best hyperparameters for each model are selected automatically with the use of Transformers library [23] and its integration with Optuna [22]. Each model trained with its best hyperparameter combination is saved for later use.

Tables 2 and 3 show the f1-score [24] for English and Spanish models, respectively, on the validation set for this first step. The best model per task and language is highlighted.

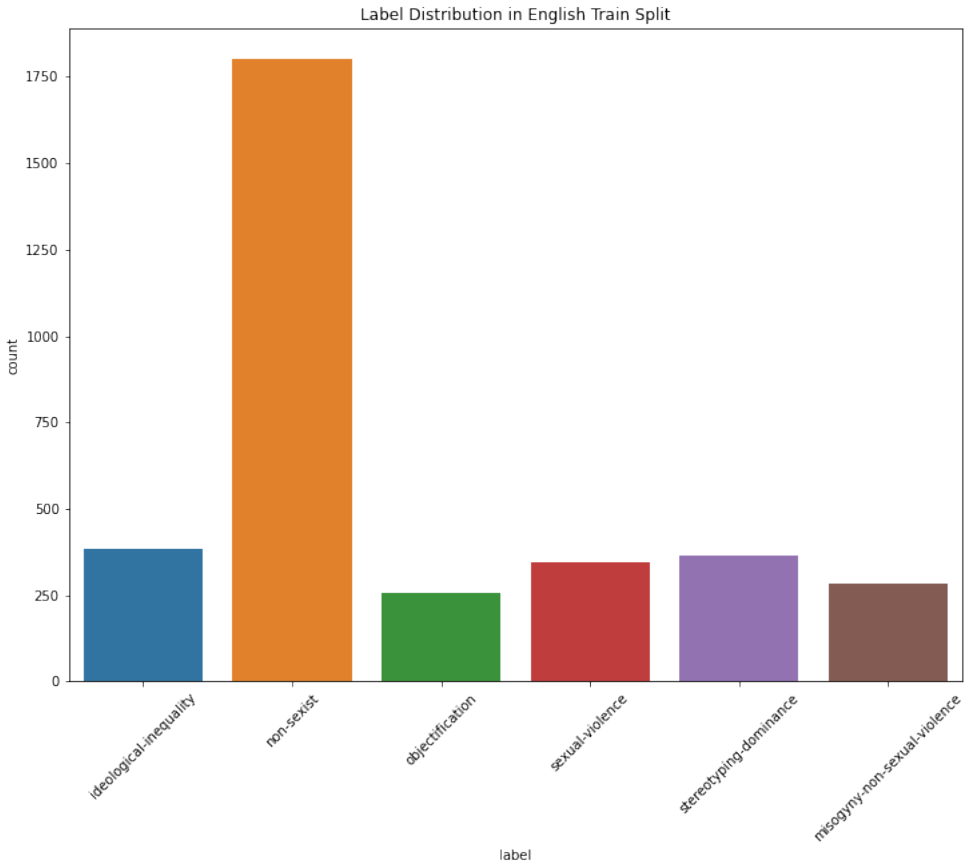

Figure 4: Number of elements per class for task 2 in English.

Hyperparameter	Values
Learning Rate	(8e-6, 8e-5, log)
Num Train Epochs	{3, 5, 7, 10, 15}
Train Batch Size	{16, 32, 48, 64, 96, 128}
Weight Decay	(0.0, 0.3)
Adam Epsilon	(1e-8, 1e-6)
Adam β_2	{0.98, 0.99}

Table 1
Hyperparameter space for tasks 1 and 2 in Spanish and English.

On both cases, and for both tasks, it should be noted that the best model is the domain-specific model: for English, it is BERTweet-large [17], while for Spanish it is RoBERTuito [21]. This makes sense since the Twitter domain is very specific, with expressions and forms of writing that are not very typical of the general domain, therefore models trained on general domain corpus may have not been exposed to this type of texts as much as the domain-specific models.

Table 2
F1-Score Results for Best Combination of Hyperparameters per Model in Tasks 1 and 2 in English.

Model	Task	F1
DeBERTa-v3-large	Task 1	0.828
RoBERTa-large	Task 1	**0.831***
BERTweet-large	Task 1	**0.831***
DeBERTa-v3-large	Task 2	0.615
RoBERTa-large	Task 2	0.609
BERTweet-large	Task 2	**0.626***

Table 3
F1-Score Results for Best Combination of Hyperparameters per Model in Tasks 1 and 2 in Spanish.

Model	Task	F1
BERTIN	Task 1	0.789
MarIA-base	Task 1	0.80
BETO	Task 1	0.808
RoBERTuito	Task 1	**0.819***
BERTIN	Task 2	0.605
MarIA-base	Task 2	0.592
BETO	Task 2	0.606
RoBERTuito	Task 2	**0.635***

This is specially relevant given the low volume of training data for this first phase. The difference between the domain specific models and the general domain ones is bigger in Task 2, which makes sense since it is a harder task and therefore differences in performance can be more accentuated. English models have in general higher f1-score than Spanish models, which is reasonable, as models used for English are large ones, while models for Spanish are base (about half the size).

4.2. Phase 2: Re-training With More Data

Once the best hyperparameters for each model are chosen, each model is trained with more data than the previous step. Up to this point, only previous year's training data was used for training. For hyperparameter tuning this is ok, since we do not want to overfit in this regard, and preferred to select hyperparameters using less data. However, to get the full potential of each of the models, in this second step models with their best hyperparameters are trained on more data, by putting train and validation splits together and re-splitting, leaving only a random 15% of data for validation purposes. As no hyperparameter decision was going to be based on this training, the validation split is only for training stopping purposes, that is, deciding when the model has stopped generalizing.

Tables 4 and 5 show results in terms of f1-score [24] for both tasks, in English and Spanish respectively. In some cases the best resulting model varies with respect to the results over the

Table 4
F1-Score Results Per Model Training With More Data in Tasks 1 and 2 in English.

Model	Task	F1
DeBERTa-v3-large	Task 1	0.856
RoBERTa-large	Task 1	0.851
BERTweet-large	Task 1	**0.903***
DeBERTa-v3-large	Task 2	**0.729***
RoBERTa-large	Task 2	0.695
BERTweet-large	Task 2	0.682

Table 5
F1-Score Results Per Model Training With More Data in Tasks 1 and 2 in Spanish.

Model	Task	F1
BERTIN	Task 1	0.88
MarIA-base	Task 1	**0.883***
BETO	Task 1	0.87
RoBERTuito	Task 1	0.863
BERTIN	Task 2	0.777
MarIA-base	Task 2	0.752
BETO	Task 2	**0.82***
RoBERTuito	Task 2	0.712

bigger validation set, seen in tables 2 and 3. This is normal since differences between models are relatively small and the use of a different, smaller validation set can significantly alter models' metrics.

In both languages it can be observed that with more training data (and less validation data), models achieve higher scores in general, thus proving our point of training with the best hyperparameters with more training data. Specially when there is scarcity for training data, as in the case of this competition, selecting a bigger proportion of it for training and less for validation can have a big impact.

4.3. Ensembling Predictions

It is known that machine learning models, in general, tend to be biased. However, when multiple models are used for predicting a new item, a less biased prediction is expected. This is the base idea underlying ensembles theory, in which multiple models are used to build a new one. In this case, we do not build a meta-model on top of the base models presented before; but an aggregating rule is developed.

For each language, all re-trained models (second phase of training explained above) are loaded, together with their validation scores. Based on these validation scores, a coefficient or weight from 0.5 to 1.0 is set for each model (these coefficients were also set in experiments from 0.7 to 1.0 but gave worse results). This is done using MinMaxScaler from Scikit-Learn [25]. A

Table 6
F1-Score for Ensemble in Tasks 1 and 2 in Spanish and English.

Language	Task	Phase	F1
ES	Task 1	Phase 1	0.81
ES	Task 1	Phase 2	0.830
ES	Task 2	Phase 1	0.636
ES	Task 2	Phase 2	0.66
EN	Task 1	Phase 1	0.80
EN	Task 1	Phase 2	0.817
EN	Task 2	Phase 1	0.594
EN	Task 2	Phase 2	0.569

dictionary with the models' validation scores is used, transforming its values to the 0.5-1.0 scale. This weight setting is automatic based on validation scores, which are obtained from log files.

Then, for prediction, logits from each of those models are obtained for the item being predicted. These logits are multiplied by the models' coefficients obtained previously. The average of those logits is computed, therefore obtaining a weighted mean of the logits for all models in each language. The label with the maximum weighted averaged logit is selected as the prediction.

This method is the final method used for both tasks, as our experiments show it worked significantly better than using the best possible model for each language and task alone.

5. Experiments

As an additional experiment, and given the good results of the simple ensembling strategy presented previously, a pure Transformers Ensemble model was implemented. This model has 3 encoders, for example 2 BERT and 1 RoBERTa, which are restricted to have the same hidden dimension size. Then, inputs are passed through each of those models, getting their last hidden state. These are concatenated, therefore producing a vector of size $3 \cdot hidden\_size$. This vector is then used to get the logits for each class, by passing it through a linear layer with input size $3 \cdot hidden\_size$ and output size n_labels. Before this linear layer, dropout is applied.

It should be noticed that for preparing inputs, each base model forming the ensemble has its own first layer, therefore their vocabularies and tokenizers don't coincide. This causes additional preprocessing, as the input ids are obtained separately for each submodel in the Ensemble from ther same original text.

This approach has several disadvantages, though. The first one is that, given that there is not much research in this regard, there are no reported recommended hyperparameters for such models, therefore we would need to carry out many experiments to find out which hyperparameters stabilize and optimize the training of such an Ensemble. Another clear disadvantage is the computing time.

Table 6 contains all results in terms of f1-score for ensembles built for tasks 1 and 2 in Spanish and English, both with phase 1 and phase 2 data splits and setup, already explained before.

Table 7
Definitive F1-Score Results for the Test Set of EXIST2022.

Task	F1	Accuracy	Position in competition
Task 1	0.7978	0.7996	1
Task 2	0.5106	0.7013	1

Although the results for the ensemble are in general close to the best performing models in each task and language, in no case it is able to clearly outperform them, in spite of the substantial increase in prediction and training time and resources usage. For this reason, this is left as an interesting experiment, but it is not used for getting the final predictions, as the simple ensembling strategy presented in subsection 4.3 is simpler, uses less resources and obtains better results.

6. Results in Test Sets

Results in different validation splits have been shown previously. Tables 2 and 3 show results for the validation set provided by the competition organizers, which correspond to the test set from 2021 edition. Tables 4 and 5 show the results after re-training, over a smaller validation set selected randomly.

In this section the final results for both tasks are presented. Table 7 show the results for the ensembling strategy on tasks 1 and 2 of the EXIST challenge.

As seen in table 7, our simple ensembling strategy obtains the best overall results on both task 1 and 2, therefore achieving the highest results of the competition. Full results can be accessed in EXIST2022's official webpage.

7. Conclusions and Future Work

In this work different solutions for tasks 1 and 2 of EXIST2022, a workshop aimed at detecting and categorizing sexism in Spanish and English, are presented. For that, a full review of both Spanish and English language models is carried out, to identify the best candidate models for each language.

Then, training is done in 2 phases. In the first phase the best hyperparameters are found for each model. In the second phase, these are used to train the models with more data used for training. Finally, a simple ensembling strategy is used for improving the systems' predictions and reducing the impact of each model's bias. This strategy is based on giving a different weight to each model, depending on its performance over a small validation set.

Pure Transformers Ensembles are presented in section 5, although they do not work yet to its full potential, arguably due to the unkown proper hyperparameter settings for this type of model. Therefore, for future work, more research in terms of appropriate hyperparameter settings and experimental setup for this type of models will be carried out.

Finally, as shown in table 7, the simple ensembling strategy presented in subsection 4.3, which uses models presented in section 2 and subsections 4.1 and 4.2, obtains the best results on both task 1 and task 2 of the EXIST2022 workshop.

References

[1] F. Rodríguez-Sánchez, J. Carrillo-de Albornoz, L. Plaza, A. Mendieta-Aragón, G. Marco-Remón, M. Makeienko, M. Plaza, J. Gonzalo, D. Spina, P. Rosso, Overview of exist 2022: sexism identification in social networks, Procesamiento del Lenguaje Natural 69 (2022).

[2] M. Anzovino, E. Fersini, P. Rosso, Automatic identification and classification of misogynistic language on twitter, in: NLDB, 2018.

[3] S. Frenda, B. Ghanem, M. Montes-y Gómez, P. Rosso, Online hate speech against women: Automatic identification of misogyny and sexism on twitter, Journal of Intelligent & Fuzzy Systems 36 (2019) 4743–4752.

[4] D. Grosz, P. Conde-Cespedes, Automatic detection of sexist statements commonly used at the workplace, 2020. arXiv:2007.04181.

[5] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. URL: https://aclanthology.org/D14-1162. doi:10.3115/v1/D14-1162.

[6] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–80. doi:10.1162/neco.1997.9.8.1735.

[7] M. Schütz, J. Boeck, D. Liakhovets, D. Slijepcevic, A. Kirchknopf, M. Hecht, J. Bogensperger, S. Schlarb, A. Schindler, M. Zeppelzauer, Automatic sexism detection with multilingual transformer models, CoRR abs/2106.04908 (2021). URL: https://arxiv.org/abs/2106.04908. arXiv:2106.04908.

[8] A. F. M. de Paula, R. F. da Silva, I. B. Schlicht, Sexism prediction in spanish and english tweets using monolingual and multilingual BERT and ensemble models, CoRR abs/2111.04551 (2021). URL: https://arxiv.org/abs/2111.04551. arXiv:2111.04551.

[9] F. Rodríguez-Sánchez, J. Carrillo-de Albornoz, L. Plaza Morales, J. Gonzalo Arroyo, P. Rosso, M. Comet, T. Donoso, Overview of exist 2021: sexism identification in social networks, 2021-09.

[10] F. Rodríguez-Sánchez, J. Carrillo-de Albornoz, L. Plaza, Automatic classification of sexism in social networks: An empirical study on twitter data, IEEE Access 8 (2020) 219563–219576. doi:10.1109/ACCESS.2020.3042604.

[11] F. Rodríguez-Sánchez, J. Carrillo-de Albornoz, L. Plaza, A multi-task and multilingual model for sexism identification in social networks, 2021. URL: http://ceur-ws.org/Vol-2943/exist_paper13.pdf.

[12] J. e. a. Devlin, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/pdf/1810.04805.pdf.

[13] Y. e. a. Liu, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/pdf/1907.11692.pdf.

[14] R. C. et al., Exploring the limits of transfer learning with a unified text-to-text transformer, 2020. URL: https://arxiv.org/pdf/1910.10683.pdf.

[15] P. e. a. He, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: https://arxiv.org/pdf/2006.03654.pdf.

[16] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, CoRR abs/2111.09543 (2021). URL: https://arxiv.org/abs/2111.09543. arXiv:2111.09543.

[17] D. Q. Nguyen, T. Vu, A. T. Nguyen, Bertweet: A pre-trained language model for english tweets, CoRR abs/2005.10200 (2020). URL: https://arxiv.org/abs/2005.10200. arXiv:2005.10200.

[18] J. e. a. Cañete, Spanish pre-trained bert model and evaluation data, 2020. URL: https://users.dcc.uchile.cl/~jperez/papers/pml4dc2020.pdf.

[19] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, A. Gonzalez-Agirre, C. Armentano-Oller, C. R. Penagos, M. Villegas, Spanish language models, CoRR abs/2107.07253 (2021). URL: https://arxiv.org/abs/2107.07253. arXiv:2107.07253.

[20] J. D. la Rosa y Eduardo G. Ponferrada y Manu Romero y Paulo Villegas y Pablo González de Prado Salas y María Grandury, Bertin: Efficient pre-training of a spanish language model using perplexity sampling, Procesamiento del Lenguaje Natural 68 (2022) 13–23. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6403.

[21] J. M. Pérez, D. A. Furman, L. A. Alemany, F. Luque, Robertuito: a pre-trained language model for social media text in spanish, CoRR abs/2111.09453 (2021). URL: https://arxiv.org/abs/2111.09453. arXiv:2111.09453.

[22] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, 2019. arXiv:1907.10902.

[23] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Huggingface's transformers: State-of-the-art natural language processing, CoRR abs/1910.03771 (2019). URL: http://arxiv.org/abs/1910.03771. arXiv:1910.03771.

[24] J. Opitz, S. Burst, Macro f1 and macro f1, 2021. arXiv:1911.03347.

[25] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.